

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. — 56. (Cancelled)

57. (New) A method, comprising:

- obtaining a circular array comprising a first data item stored in a first location, a second data item stored in a second location immediately following the first location, and a third data item stored in a third location immediately following the second location;

- obtaining a front counter indexing a front location immediately following the third location and a back counter indexing a back location immediately before the first location;

- executing a first pop back instruction using a first double compare and swap (DCAS) operation to remove the first data item from the circular array, place a null value in the first location, and change the back counter to index the first location;

- executing a first pop front instruction using a second DCAS operation to remove the third data item from the circular array, place the null value in the third location of the circular array, and change the front counter to index the third location; and

- executing a second pop front instruction without reading the back counter after executing the first pop back instruction and the first pop front instruction.

58. (New) The method of claim 57, further comprising:

- storing at least one selected from a group consisting of the first data item and the second data item after executing the first pop back instruction and the first pop front instruction.

59. (New) The method of claim 57, wherein executing the second pop front instruction comprises:

- determining a first counter value by reading the front counter;

- calculating a second counter value based on the first counter value; and

- determining a first array value by reading the circular array at the second counter value.

60. (New) The method of claim 59, wherein executing the second pop front instruction further comprises:

- comparing the first counter value with the front counter to generate a first comparison;
- and
- comparing the first array value with a second array value determined by reading the circular array at the second counter value to generate a second comparison,
- wherein the first array value equals the second array value and the first counter value equals the second counter value,
- wherein the first comparison and the second comparison belong to a third DCAS operation executed after determining the first array value, and
- wherein the third DCAS operation removes the second data item from the circular array and sets the front counter to index the second location based on the first comparison and the second comparison.

61. (New) The method of claim 59, further comprising:

- executing a second pop back instruction after determining the first array value to remove the second data item from the circular array, place the null value in the second location of the circular array, and change the back counter to index the second location;
- comparing the first counter value with the front counter to generate a first comparison;
- and
- comparing the first array value with a second array value determined by reading the circular array at the second counter value to generate a second comparison,
- wherein the first comparison and the second comparison belong to a third DCAS operation executed after executing the second pop back instruction, and
- wherein the third DCAS indicates the circular array is empty based on the first comparison and the second comparison.

62. (New) The method of claim 59, further comprising:

- executing a push front instruction after determining the first array value to add a fourth data item to the third position and change the front counter to index the front location;

comparing the first counter value with the front counter to generate a first comparison;
comparing the first array value with a second array value determined by reading the
circular array at the second counter value to generate a second comparison,
wherein the first comparison and the second comparison belong to a third DCAS
operation executed after executing the push front instruction; and
returning the fourth data item from the circular array and setting the front counter to
index the third location using a fourth DCAS after executing the third DCAS.

63. (New) The method of claim 57, wherein the second pop front instruction is issued by a first thread and the second pop back instruction is issued by a second thread.
64. (New) The method of claim 57, wherein the second pop front instruction is issued by a first processor and the second pop back instruction is issued by a first thread.
65. (New) The method of claim 59, wherein the second counter value equals the first counter value minus one modulo a size of the circular array.
66. (New) The method of claim 59, wherein the second counter value equals the first counter value plus one modulo a size of the circular array.
67. (New) The method of claim 60, wherein the first counter value equals the front counter.
68. (New) The method of claim 61, wherein the second array value is null and the first counter value equals the front counter.
69. (New) The method of claim 61, wherein the second pop front instruction is issued by a first thread and the second pop back instruction is issued by a second thread.
70. (New) The method of claim 62, wherein an absolute difference between the first counter value and the front counter exceeds zero.
71. (New) The method of claim 62, wherein the push front instruction is issued by a first thread and the second pop front instruction is issued by a second thread.

72. (New) A method, comprising:

- obtaining a circular array comprising a plurality of data items, wherein each of the plurality of data items is stored in one of a plurality of consecutive locations in the circular array, wherein a size of the circular array exceeds a cardinality of the plurality of data items by at least three locations;
- obtaining a front counter indexing a front location of the circular array immediately following the plurality of consecutive locations and a back counter indexing a back location of the circular array immediately before plurality of consecutive locations;
- executing a first push back instruction using a first double compare and swap (DCAS) operation to insert a first data item into the back location and change the back counter to index an adjacent location immediately before the back location;
- executing a first push front instruction using a second DCAS operation to insert a second data item into the front location and change the front counter to index the adjacent location; and
- executing a second push front instruction without reading the back counter after executing the first push back instruction and the first push front instruction.

73. (New) The method of claim 72, further comprising:

- storing at least one selected from a group consisting of the first data item and the second data item after executing the first push back instruction and the first push front instruction.

74. (New) The method of claim 72, wherein executing the second push front instruction comprises:

- determining a first counter value by reading the front counter;
- calculating a second counter value based on the first counter value; and
- determining a first array value by reading the circular array at the first counter value.

75. (New) The method of claim 74, wherein executing the second push front instruction further comprises:

- comparing the first counter value with the front counter to generate a first comparison;
- and
- comparing the first array value with a second array value determined by reading the circular array at the first counter value to generate a second comparison,

wherein the first array value equals the second array value and the first counter value equals the second counter value,

wherein the first comparison and the second comparison belong to a third DCAS operation executed after determining the first array value, and

wherein the third DCAS operation inserts a third data item into the adjacent location and sets the front counter to the second counter value based on the first comparison and the second comparison.

76. (New) The method of claim 74, further comprising:

- executing a second push back instruction after determining the first array value to insert a fourth data item into the adjacent location and change the back counter to index the front location;
- comparing the first counter value with the front counter to generate a first comparison;
- and
- comparing the first array value with a second array value determined by reading the circular array at the first counter value to generate a second comparison,

wherein the first comparison and the second comparison belong to a third DCAS operation executed after executing the second push back instruction, and

wherein the third DCAS indicates the circular array is full based on the first comparison and the second comparison.

77. (New) The method of claim 74, further comprising:

- executing a pop front instruction after determining the first array value to remove the second data item and change the front counter to index the front location;
- comparing the first counter value with the front counter to generate a first comparison;

comparing the first array value with a second array value determined by reading the circular array at the first counter value to generate a second comparison, wherein the first comparison and the second comparison belong to a third DCAS operation executed after executing the pop front instruction; and inserting a third data item into the front location and setting the front counter to index the adjacent location using a fourth DCAS after executing the third DCAS.

78. (New) The method of claim 72, wherein the first push front instruction is issued by a first thread and the second push front instruction is issued by a second thread.
79. (New) The method of claim 74, wherein the second counter value equals the first counter minus one modulo a size of the circular array.
80. (New) The method of claim 74, wherein the second counter value equals the first counter value plus one modulo a size of the circular array.
81. (New) The method of claim 75, wherein the first array value is null and the first counter value equals the front counter.
82. (New) The method of claim 76, wherein the first counter value equals the front counter.
83. (New) The method of claim 76, wherein the second push back instruction is issued by a first thread and the second push front instruction is issued by a second thread.
84. (New) The method of claim 77, wherein an absolute difference between the first counter value and the front counter exceeds zero.
85. (New) The method of claim 77, wherein the pop front instruction is issued by a first thread and the second push front instruction is issued by a second thread.
86. (New) An apparatus, comprising:
means for obtaining a circular array comprising a first data item stored in a first location, a second data item stored in a second location immediately following the first location, and a third data item stored in a third location immediately following the second location;

- means for obtaining a front counter indexing a front location immediately following the third location and a back counter indexing a back location immediately before the first location;
- means for executing a first pop back instruction using a first double compare and swap (DCAS) operation to remove the first data item from the circular array, place a null value in the first location, and change the back counter to index the first location;
- means for executing a first pop front instruction using a second DCAS operation to remove the third data item from the circular array, place the null value in the third location of the circular array, and change the front counter to index the third location; and
- means for executing a second pop front instruction without reading the back counter after executing the first pop back instruction and the first pop front instruction.

87. (New) The apparatus of claim 86, wherein the means for executing the second pop front instruction comprises:

- means for determining a first counter value by reading the front counter;
- means for calculating a second counter value based on the first counter value; and
- means for determining a first array value by reading the circular array at the second counter value.

88. (New) The apparatus of claim 87, wherein the means for executing the second pop front instruction further comprises:

- means for comparing the first counter value with the front counter to generate a first comparison; and
 - means for comparing the first array value with a second array value determined by reading the circular array at the second counter value to generate a second comparison,
- wherein the first array value equals the second array value and the first counter value equals the second counter value,
- wherein the first comparison and the second comparison belong to a third DCAS operation executed after determining the first array value, and

wherein the third DCAS operation removes the second data item from the circular array and sets the front counter to index the second location based on the first comparison and the second comparison.

89. (New) The apparatus of claim 87, further comprising:

means for executing a second pop back instruction after determining the first array value to remove the second data item from the circular array, place the null value in the second location of the circular array, and change the back counter to index the second location;

means for comparing the first counter value with the front counter to generate a first comparison; and

means for comparing the first array value with a second array value determined by reading the circular array at the second counter value to generate a second comparison,

wherein the first comparison and the second comparison belong to a third DCAS operation executed after executing the second pop back instruction, and

wherein the third DCAS indicates the circular array is empty based on the first comparison and the second comparison.

90. (New) The apparatus of claim 87, further comprising:

means for executing a push front instruction after determining the first array value to add a fourth data item to the third position and change the front counter to index the front location;

means for comparing the first counter value with the front counter to generate a first comparison;

means for comparing the first array value with a second array value determined by reading the circular array at the second counter value to generate a second comparison,

wherein the first comparison and the second comparison belong to a third DCAS operation executed after executing the push front instruction; and

means for returning the fourth data item from the circular array and setting the front counter to index the third location using a fourth DCAS after executing the third DCAS.

91. (New) An apparatus comprising:

means for obtaining a circular array comprising a plurality of data items, wherein each of the plurality of data items is stored in one of a plurality of consecutive locations in the circular array, wherein a size of the circular array exceeds a cardinality of the plurality of data items by at least three locations;

means for obtaining a front counter indexing a front location of the circular array immediately following the plurality of consecutive locations and a back counter indexing a back location of the circular array immediately before plurality of consecutive locations;

means for executing a first push back instruction using a first double compare and swap (DCAS) operation to insert a first data item into the back location and change the back counter to index an adjacent location immediately before the back location;

means for executing a first push front instruction using a second DCAS operation to insert a second data item into the front location and change the front counter to index the adjacent location; and

means for executing a second push front instruction without reading the back counter after executing the first push back instruction and the first push front instruction.

92. (New) The apparatus of claim 91, wherein the means for executing the second push front instruction comprises:

means for determining a first counter value by reading the front counter;

means for calculating a second counter value based on the first counter value; and

means for determining a first array value by reading the circular array at the first counter value.

93. (New) The apparatus of claim 92, wherein the means for executing the second push front instruction further comprises:

- means for comparing the first counter value with the front counter to generate a first comparison; and
- means for comparing the first array value with a second array value determined by reading the circular array at the first counter value to generate a second comparison,
- wherein the first array value equals the second array value and the first counter value equals the second counter value,
- wherein the first comparison and the second comparison belong to a third DCAS operation executed after determining the first array value, and
- wherein the third DCAS operation inserts a third data item into the adjacent location and sets the front counter to the second counter value based on the first comparison and the second comparison.

94. (New) The apparatus of claim 92, further comprising:

- means for executing a second push back instruction after determining the first array value to insert a fourth data item into the adjacent location and change the back counter to index the front location;
- means for comparing the first counter value with the front counter to generate a first comparison; and
- means for comparing the first array value with a second array value determined by reading the circular array at the first counter value to generate a second comparison,
- wherein the first comparison and the second comparison belong to a third DCAS operation executed after executing the second push back instruction, and
- wherein the third DCAS indicates the circular array is full based on the first comparison and the second comparison.

95. (New) The apparatus of claim 92, further comprising:

- means for executing a pop front instruction after determining the first array value to remove the second data item and change the front counter to index the front location;
- means for comparing the first counter value with the front counter to generate a first comparison;
- means for comparing the first array value with a second array value determined by reading the circular array at the first counter value to generate a second comparison,
- wherein the first comparison and the second comparison belong to a third DCAS operation executed after executing the pop front instruction; and
- means for inserting a third data item into the front location and setting the front counter to index the adjacent location using a fourth DCAS after executing the third DCAS.